Fast Fractal Image Decoding Based on Minimum Iterated Function System

Wang Qiang*

College of Information Science and Technology, Dalian Maritime University

Linghai Road, No.1, Dalian, China, 116026

*Corresponding Author, E-mail: wangqiang2011@dlmu.edu.cn

Abstract: To accelerate the fractal decoding process, a minimum iterated function system based fast fractal decoding method was proposed in this study. In fractal encoding process, we found that there exists a minimum domain block set (MDBS) which can provide the best-matched domain blocks for all range blocks, and then the domain blocks of MDBS, the range blocks within MDBS, and the associated mapping operations between them constitute the minimum iteration function system (MIFS). In decoding process, MIFS was first recovered in the first iteration. Then, in each of the second to penultimate iterations, only the range blocks within MDBS are reconstructed, and the computations of reconstructing the remaining range blocks can be saved. Finally, all range blocks are reconstructed to obtain the decoded image in the last iteration. Four fractal encoding methods were adopted to assess the performance of the proposed method. Experimental results show that the proposed method can complete the decoding process with fewer computations.

Keywords: Fractal image decoding; Minimum iterated function system; Minimum domain block set; Redundant domain block set; fixed point

1. Introduction

The kernel idea of fractal image coding was first proposed by Barnsley, and then Jacquin proposed the first practical fractal coding algorithm which was considered as a promising image compression method[1,2]. Although suffering from high computational complexity in encoding process, fractal image coding has the characteristics of novel idea, potential high compression ratio, fast decoding, and resolution independence. Thus, on one hand, to accelerate the encoding process, researchers worldwide proposed all kinds of fast fractal encoding methods which can be mainly divided into two categories: 1) Local block matching based fast fractal image coding[3-5]. By converting exhaustive block matching into local block matching in domain block pool, this type of methods tried to accelerate the encoding process

while maintaining the decoded image quality as much as possible. 2) No-search fractal image coding[6-9]. This type of methods appointed the best-matched domain block directly without block matching operations and can realize real-time encoding at the expense of poor decoded image quality. On the other hand, after many years' research, fractal image coding has been gradually applied to many other image processing fields, such as image denoising[10-16], image magnification[17-21], image hashing[22,23], image retrieval[24-26], watermarking[27-29], and human pose estimation[30,31].

In our study, a novel method was proposed to accelerate the fractal decoding process. Fractal encoding process generally consists of a series of mapping operations from domain blocks to range blocks. First, we found that the block matching operations can be completed only with partial domain blocks which are defined as the minimum domain block set (MDBS). Then, the remaining domain blocks outside MDBS are defined as the redundant domain block set (RDBS). Further, the domain blocks of MDBS, the range blocks within MDBS, and the associated mapping operations are defined as the minimum iterated function system (MIFS). From the above definitions, we know that by excluding the domain blocks of RDBS, MIFS contains necessary mappings operations from MDBS to all range blocks and plays a crucial role in encoding process. In decoding process, in first iteration, all range blocks are reconstructed. By distinguishing MDBS from RDBS, MIFS can be recovered. In each of the second to penultimate iterations, only the range blocks within MDBS are reconstructed by MIFS. In the last iteration, the local image within MDBS converges, and then the range blocks within RDBS are reconstructed as well. In summary, in the first and last iterations, we perform the same operations with the previous method. But for each of the second to penultimate iterations, because the domain blocks within RDBS provide no mapping operation for any range block, there is no need to reconstruct the range blocks within RDBS, and the associated computations can be saved. Three state of the art methods and Jacquin's method are adopted to assess the performance of the proposed method. Experimental results show that compared with the conventional method, the proposed method can effectively reduce the number of computations in decoding process.

2. Conventional fractal image coding

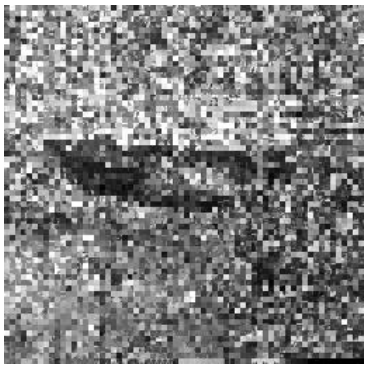
Conventional fractal image encoding proposed by Jacquin can be described in detail as[2]:

The $M \times N$ input image is first divided into nonoverlapping $B \times B$ range blocks, $\boldsymbol{R}_i, i = 1,2,3,...,\mathrm{NumR}$, where NumR denotes the total number of range blocks. Then, the domain blocks, $\boldsymbol{D}_j, j = 1,2,3,...,\mathrm{NumD}$, can be obtained by sliding a $2B \times 2B$ window over the input image with a fixed sliding step $\delta=2B$, where NumD denotes the total number of domain blocks. Further, all domain blocks are contracted to the size of $B \times B$ and constitute a domain block pool which is then extended with eight isometric transformations. Finally, for arbitrary range block $\boldsymbol{R}_i$, the best-matched domain block $\boldsymbol{D}_j$ can be found by minimizing the following function:

$$\mathrm{CE}(\boldsymbol{R}_i) = \underset{\boldsymbol{D}_j}{\operatorname{argmin}} \left\| \boldsymbol{R}_i - \varphi_{ij}(\boldsymbol{D}_j) \right\| = \underset{\boldsymbol{D}_j}{\operatorname{argmin}} \left\| \boldsymbol{R}_i - \left\{ \alpha_i \gamma_i \left[\eta_i (\boldsymbol{D}_j) \right] + \beta_i \boldsymbol{I} \right\} \right\| \quad (1)$$
$$i = 1,2,3,...,\mathrm{NumR},\ j = 1,2,3,...,\mathrm{NumD}$$

where $\varphi_{ij}(\cdot)$ denotes the mapping operation from $\boldsymbol{D}_j$ to $\boldsymbol{R}_i$, $\eta_i(\cdot)$ and $\gamma_i(\cdot)$ denote the contracting and isometric transformations, respectively, and α_i and β_i denote the scaling and offset coefficients of the affine transformation, respectively. $\boldsymbol{I}$ denotes the $B \times B$ block whose components are all ones.

(a) Initial Lake image



(b) First iteration

(c) Second iteration

(d) Third iteration

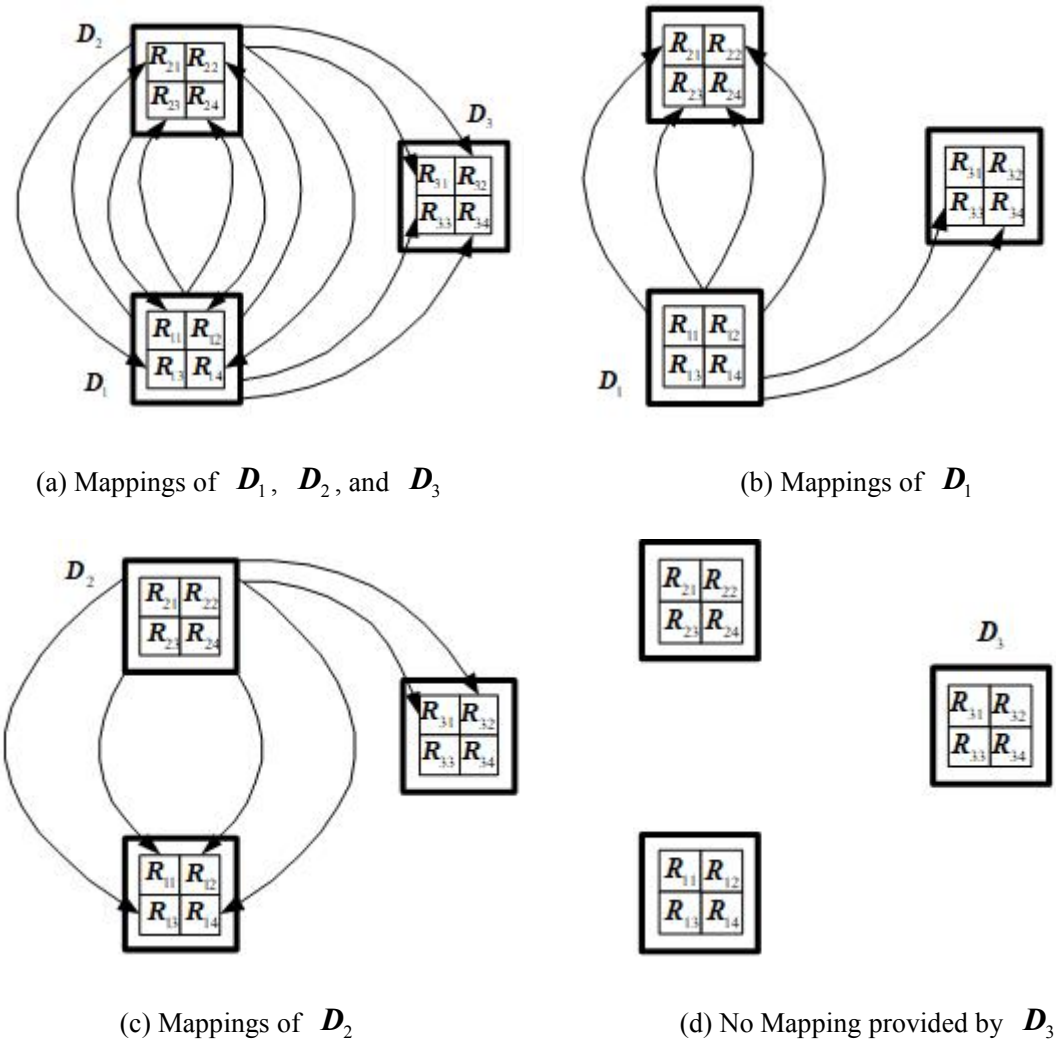
(e) Fourth iteration

(f) Fifth iteration

Fig. 1 Initial image and the first five iteration images in decoding process.

In decoding process, arbitrary $M \times N$ image can be selected as the initial image. In each of the following iterations, based on the same operations in encoding process, all range blocks are reconstructed by their respective best-matched domain blocks. After about ten iterations, the decoding process converges to the decoded image. When the Bridge and Lake images are selected as the input image in encoding process and initial image in decoding process, respectively, Figure 1 (a) illustrates the initial image, and the remaining five images in Fig.1 (b)-(f) illustrate the first five iteration images in decoding process. We can see that the initial image can gradually converge to the input image.

3. Accelerating fractal decoding process by minimum iterated function system



(a) Mappings of $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$

(b) Mappings of $\boldsymbol{D}_1$

(c) Mappings of $\boldsymbol{D}_2$

(d) No Mapping provided by $\boldsymbol{D}_3$

Fig. 2 Illustration of mapping operations from domain blocks to range blocks.

In encoding process, by a series of contracting, isometric, and affine transformations, the

range blocks can be approximated by their respective best-matched domain blocks as:

$$\boldsymbol{I}=\bigcup_{i=1}^{\mathrm{NumR}}\boldsymbol{R}_i\approx\bigcup_{i=1}^{\mathrm{NumR}}\varphi_{ij}\left(\boldsymbol{D}_j'\right)=\bigcup_{i=1}^{\mathrm{NumR}}\left\{\alpha_i\gamma_i\left[\eta_i\left(\boldsymbol{D}_j'\right)\right]+\beta_i\boldsymbol{I}\right\} \tag{2}$$
$$i=1,2,3,...,\mathrm{NumR},\ j=1,2,3,...,\mathrm{NumD}'$$

where $\boldsymbol{D}_j'$ denotes the best-matched domain block of $\boldsymbol{R}_i$, NumD' denotes total number of the best-matched domain blocks we actually need and satisfies $\mathrm{NumD}'\le\mathrm{NumD}$. Although all domain blocks, $\boldsymbol{D}_j, j=1,2,3,...,\mathrm{NumD}$, are the candidates of the best-matched domain block of $\boldsymbol{R}_i$, only partial domain blocks, $\boldsymbol{D}_j', j=1,2,3,...,\mathrm{NumD}'$, are the ones we actually need. The remaining domain blocks cannot provide mapping operation for any range block and will be not used at all. For example, in Fig.2 (a), we have three domain blocks, $\boldsymbol{D}_j, j=1,2,3$, and each domain block can be uniformly divided into four range blocks, $\boldsymbol{R}_{ji}, i=1,2,3,4$. The arrow "$\rightarrow$" represents the mapping operation $\varphi_{ij}(\cdot)$ from one domain block to another range block. Figure 2 (b), (c), and (d) illustrate the mapping operations that are provided by domain blocks $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$, respectively. We can see that in Fig.2 (b) and (c), both $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ can provide mapping operations for range blocks. But in Fig.2 (d), $\boldsymbol{D}_3$ provides no mapping operation for any range block. Thus, we can divide the domain blocks into two categories: Minimum domain block set (MDBS) and redundant domain block set (RDBS):

Definition 1: The domain blocks that can provide the mapping operations for range blocks more than or equal to once constitute the minimum domain block set (MDBS):

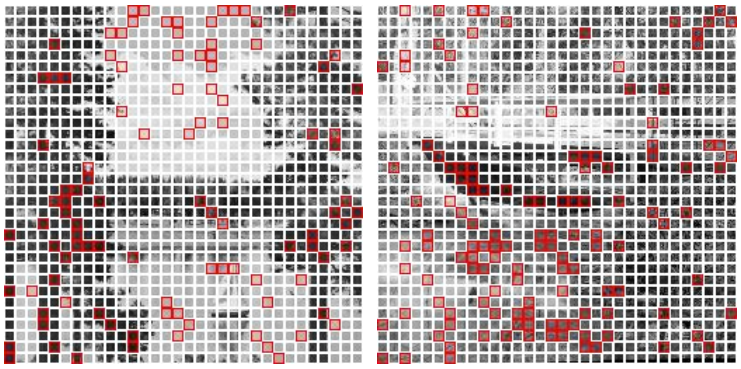
$$\mathrm{MDBS}=\bigcup_j\boldsymbol{D}'\ \text{s.t.}\ \mathrm{Times}\left(\boldsymbol{D}_j',\boldsymbol{R}_i\right)\ge 1, i=1,2,3,...,\mathrm{NumR},\ j=1,2,3,...,\mathrm{NumD}' \tag{3}$$

where Times(A, B) denotes the total number of mapping operations from A to B. From Definition 1, we know that the domain blocks of MDBS can provide mapping operations for all range blocks (including the range blocks within MDBS), In Fig.2, both $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ represent the domain blocks of MDBS, and then we have the definition of redundant domain block set (RDBS) as:

Definition 2: The domain blocks that provide no mapping operation for any range block constitute the redundant domain block set (RDBS):

$$\mathrm{RDBS}=\bigcup_{j}\boldsymbol{D}_{j}'' \text{ s.t. Times}\left(\boldsymbol{D}_{j}'',\boldsymbol{R}_{i}\right)=0,\ i=1,2,3,...,\mathrm{NumR},\ j=1,2,3,...,\mathrm{NumD}'' \tag{4}$$

where NumD'' denote the total number of the domain blocks of RDBS and satisfies $\mathrm{NumD}''=\mathrm{NumD}-\mathrm{NumD}'$. In Fig.2, $\boldsymbol{D}_3$ represents the domain block of RDBS. For Lake and Bridge images in Fig.3 (a) and (b), the domain blocks within the white and red boxes represent the domain blocks of MDBS and RDBS, respectively.



(a)Lake (b) Bridge

Fig.3 Illustration of MDBS and RDBS.

Further, we have the definition of minimum iterated function system (MIFS) as:

Definition 3: The domain blocks of MDBS, the range blocks within MDBS, and the associated mapping operations between them are defined as the minimum iterated function system (MIFS):

$$\mathrm{MIFS}:\left[\left(\boldsymbol{R}_{i}',\boldsymbol{D}_{j}'\right),\varphi_{ij}\right]\quad i=1,2,3,...,\mathrm{NumR}',\ j=1,2,3,...,\mathrm{NumD}',\ \boldsymbol{D}_{j}'\in\mathrm{MDB\ S} \tag{5}$$

where $\boldsymbol{R}_{i}', i=1,2,3,...,\mathrm{NumR}'$ denote the range blocks within MDBS. NumR' denotes the total number of the range blocks within MDBS and satisfies $\mathrm{NumR}'=\mathrm{NumD}'\times 4$. From Definition 1, 2, and 3, we know that because MIFS can provide mapping operations from the domain blocks of MDBS to the range blocks within MDBS, MIFS can complete the mapping operations by itself independently. Moreover, because the domain blocks of MDBS within MIFS can provide mapping operations for the range blocks within RDBS as well, MIFS can be used to reconstruct the whole decoded image in decoding process. Figure 4 illustrates the MIFS contained in Fig.2 (a).

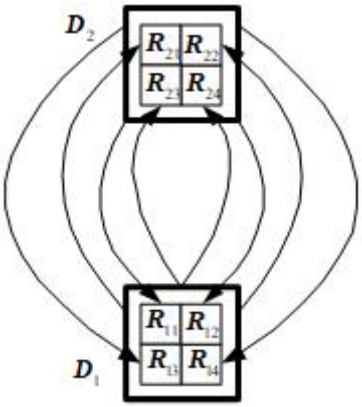


Fig.4 MIFS contained in Fig.2

In decoding process, the proposed decoding method can be designed as: In first iteration, based on the mapping operations in encoding process, we reconstruct all range blocks, and MIFS can be recovered by distinguishing MDBS from RDBS. Then, in each of the second to penultimate iterations, based on MIFS, only the range blocks within MDBS are reconstructed by the domain blocks of MDBS. On one hand, for the range blocks within MDBS, MIFS maintains the same reconstruction operations as the previous method. Thus, we have the same local iteration image within MDBS as before which can be used to judge whether convergence has been achieved, and the same number of iterations can be also maintained. On the other hand, based on Definition 2, because the domain blocks of RDBS do not take part in reconstructing any range block, there is no need to reconstruct the range blocks within RDBS, and the associated computations can be saved. Finally, in the last iteration, after the range blocks within MDBS are reconstructed, the range blocks within RDBS are reconstructed as well, and the whole decoded image can be obtained. Figure 5 illustrates the process of the proposed decoding method, and we can see that the proposed method maintain the same number of iterations as the previous method. In the first and last iterations, the unshaded regions represent all range blocks within MDBSs and RDBSs we need to reconstruct, and the proposed method does the same operations with the previous method. For each of the second to penultimate iterations, the unshaded regions still represent the range blocks within MDBSs we need to reconstruct. However, the shadow regions represent the range blocks within RDBSs whose reconstruction operations can be saved, and thus the decoding process can be accelerated.

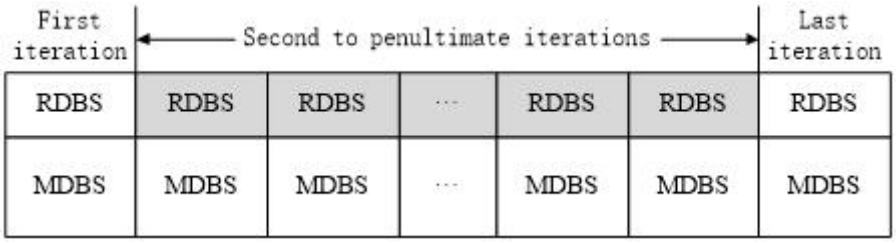


Fig.5 Flow chart of the proposed decoding process.

4. Experiments

In this section, two 256 × 256 images, Lake and Bridge, are selected as the test images. The size of range blocks is set at 4 × 4, and the sliding step is 8. The scaling and offset coefficients, *s* and *o*, are quantized with 5 and 7 bits, respectively. Jacquin's method and three state of the art methods, Jacquin's[2], Chauraisa's[3], Zheng's[4], and Gupta's methods[5], are adopted to assess the performance of the proposed method. The root of mean square error (RMSE) is adopted to measure the deviation between the *k*th iteration and decoded images as:

$$\mathrm{RMSE}^{(k)}=\sqrt{\frac{1}{H\times W}\sum_{j=1}^{W}\sum_{i=1}^{H}\left[\boldsymbol{f}^{(k)}(i,j)-\boldsymbol{f}^{\mathrm{Decoded}}(i,j)\right]^2}\qquad k=1,2,...,N \tag{6}$$

where $\boldsymbol{f}^{(k)}$ and $\boldsymbol{f}^{\mathrm{Decoded}}$ denote the *k*th iteration and decoded images, respectively, *H* and *W* are the height and width of the images, respectively, and *N* denotes the total number of iterations we need to complete the decoding process. In the experiment, the decoded image can be obtained by encoding and decoding the input image in advance, and the detailed experimental procedures are listed as:

Step 1: Select one fractal encoding method and encode the input image.

Step 2: In the first iteration of decoding process, select a 256 × 256 blank image as the initial image, reconstruct all range blocks, and all domain blocks can be divided into MDBS and RDBS. Let k=1, calculate and record $\mathrm{RMSE}^{(k)}$ within MDBS.

Step 3: Reconstruct the range blocks within MDBS. Let k=k+1, calculate and record $\mathrm{RMSE}^{(k)}$ within MDBS. If the local image within MDBS satisfies the convergence requirement of $\left|\mathrm{RMSE}^{(k)}-\mathrm{RMSE}^{(k-1)}\right|\le 1$, go to Step 4. If not, turn back to Step 3, and perform another iteration.

Step 4: Reconstruct the range blocks within RDBS, and the final decoded image can be obtained.

In Tables 1, 2, 3, and 4, "▲" represents achieving convergence in the Nth iteration. For Lake image in Tab.1, we totally have NumD=1024 domain blocks, and MDBS and RDBS contain NumD′=922 and NumD''=102 domain blocks which comprise $\text{Num}D'/\text{NumD}\times 100\%=90.04\%$ and $\text{Num}D''/\text{NumD}\times 100\%=9.96\%$ of the input image, respectively. In each of the second to (N-1)th iterations, we only reconstruct the range blocks within MDBS. On one hand, the range blocks within MDBS are reconstructed by the domain blocks of MDBS, which implies that the same reconstructing operations are performed within MDBS as Jacquin's method. Thus, the proposed method can keep the same RMSEs within MDBS as Jacquin's method, and the same number of iterations can be also maintained as Jacquin's method. Then, when the deviation between the current and previous local iteration images within MDBS achieves the requirement of $\left|\text{RMSE}^{(k)}-\text{RMSE}^{(k-1)}\right|\le 1$, the decoding process can be considered as convergence. On the other hand, because the proposed method only reconstruct the range blocks within MDBS rather than all range blocks in Jacquin's method, 9.96% of total computations can be saved regarding Jacquin's method. Finally, all the range blocks within both MDBS and RDBS are reconstructed in the last iteration. For the whole decoding process, the proposed method does the same with Jacquin's method in the first and last iterations. For each of the second to (N-1)th iterations, only the range blocks within MDBS are reconstructed. In each iteration, if the percentage of the computations required (PCR) for reconstructing all range blocks is considered as 100%, the PCR for the proposed method in the whole decoding process can be calculated as:

$$\text{PCR}=\frac{(N-2)\times\left(\text{Num}D'/\text{NumD}\right)\times 100\%+2\times 100\%}{N\times 100\%}\times 100\% \tag{7}$$

where the numerator and denominator represent the computations required in the proposed and previous methods, respectively. Based on Eq. (7), we know that for Lake image in Table 1, we have $\text{Num}D'/\text{NumD}\times 100\%=90.04\%$, and N=11. Thus, PCR=91.85%, and 8.15% of total computations can be saved regarding Jacquin's method. Similarly, for Bridge image, MDBS and RDBS contain NumD′=893 and NumD''=131 domain blocks, respectively, and N=11. By Eq. (7), we have PCR=89.54%, and 10.46% of total computations can be saved.

Tables 2, 3, and 4 list the experimental results for Chaurasia's, Zheng's, and Gupta's methods, respectively. In Table 2, for Lake and Bridge images, 81.35% and 79.98% of total computations are required in each of the second to ninth iterations, and PCRs are 84.74% and 83.62%, respectively. In Table 3, for the two test images, 90.14% and 87.21% of the total computations are required in each of the second to tenth iterations, and PCRs are 91.93% and 89.54%, respectively. In Table 4, for the two test images, 90.72% and 88.09% of total computations are required in each of the second to tenth iterations, and PCRs are 92.41% and 90.26%, respectively. In summary, compared with the previous methods, the proposed method can effectively reduce the number of computations in decoding process.

Table 1 Performance comparison between Jacquin's and proposed methods in decoding process[2].

Image	Methods	Performance	Iterations (k)											PCR
			1	2	3	4	5	6	7	8	9	10	11	(%)
Lake	Jacquin's	RMSE(MDBS)	97.18	67.50	46.56	31.01	20.33	13.21	8.52	5.45	3.44	2.14	1.29▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	97.18	67.50	46.56	31.01	20.33	13.21	8.52	5.45	3.44	2.14	1.29▲	×
		PCR(%)	100	**90.04**	**90.04**	**90.04**	**90.04**	**90.04**	**90.04**	**90.04**	**90.04**	**90.04**	100	**91.85**
Bridge	Jacquin's	RMSE(MDBS)	81.66	55.01	37.56	24.95	16.62	11.02	7.26	4.74	3.06	1.94	1.19▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	81.66	55.01	37.56	24.95	16.62	11.02	7.26	4.74	3.06	1.94	1.19▲	×
		PCR(%)	100	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**100**	**89.54**

Table 2 Performance comparison between Chaurasia's method and the proposed method[3].

Image	Methods	Performance	Iterations (k)										PCR
			1	2	3	4	5	6	7	8	9	10	(%)
Lake	Chaurasia's	RMSE(MDBS)	85.59	54.51	34.66	21.46	13.08	7.87	4.70	2.79	1.64	0.95▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	85.59	54.51	34.66	21.46	13.08	7.87	4.70	2.79	1.64	0.95▲	×
		PCR(%)	100	**81.35**	**81.35**	**81.35**	**81.35**	**81.35**	**81.35**	**81.35**	**81.35**	100	**84.74**
Bridge	Chaurasia's	RMSE(MDBS)	78.82	50.92	32.95	20.22	12.15	7.25	4.28	2.51	1.45	0.83▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	78.82	50.92	32.95	20.22	12.15	7.25	4.28	2.51	1.45	0.83▲	×
		PCR(%)	100	**79.98**	**79.98**	**79.98**	**79.98**	**79.98**	**79.98**	**79.98**	**79.98**	100	**83.62**

Table 3 Performance comparison between Zheng's method and the proposed method[4].

Image	Methods	Performance	Iterations (k)											PCR
			1	2	3	4	5	6	7	8	9	10	11	(%)
Lake	Zheng's	RMSE(MDBS)	97.38	67.76	46.81	31.24	20.51	13.35	8.62	5.52	3.49	2.17	1.31▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100	100

	Proposed	RMSE(MDBS)	97.38	67.76	46.81	31.24	20.51	13.35	8.62	5.52	3.49	2.17	1.31▲	×
		PCR(%)	100	**90.14**	**90.14**	**90.14**	**90.14**	**90.14**	**90.14**	**90.14**	**90.14**	**90.14**	100	**91.93**
Bridge	Zheng's	RMSE(MDBS)	81.75	54.94	37.24	24.52	16.21	10.70	7.03	4.57	2.94	1.85	1.14▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	81.75	54.94	37.24	24.52	16.21	10.70	7.03	4.57	2.94	1.85	1.14▲	×
		PCR(%)	100	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	**87.21**	100	**89.54**

Table 4 Performance comparison between Gupta's method and the proposed method[5].

Image	Methods	Performance	Iterations (k)										PCR
			1	2	3	4	5	6	7	8	9	10	11	(%)
Lake	Gupta's	RMSE(MDBS)	98.55	69.49	48.59	32.63	21.60	14.16	9.22	5.95	3.80	2.38	1.45▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	98.55	69.49	48.59	32.63	21.60	14.16	9.22	5.95	3.80	2.38	1.45▲	×
		PCR(%)	100	**90.72**	**90.72**	**90.72**	**90.72**	**90.72**	**90.72**	**90.72**	**90.72**	**90.72**	100	**92.41**
Bridge	Gupta's	RMSE(MDBS)	81.10	54.62	37.22	24.78	16.61	11.15	7.42	4.89	3.17	2.02	1.24▲	×
		PCR(%)	100	100	100	100	100	100	100	100	100	100	100	100
	Proposed	RMSE(MDBS)	81.10	54.62	37.22	24.78	16.61	11.15	7.42	4.89	3.17	2.02	1.24▲	×
		PCR(%)	100	**88.09**	**88.09**	**88.09**	**88.09**	**88.09**	**88.09**	**88.09**	**88.09**	**88.09**	100	**90.26**

5. Conclusion

A novel fast fractal decoding method is proposed in this paper. In encoding process, the definitions of MDBS, RDBS, and MIFS are introduced. In decoding process, through the same number of iterations as before, MIFS can complete the decoding process and achieve convergence within MDBS independently, and thus the computations of reconstructing the range blocks within RDBS from the second to penultimate iterations can be saved. Moreover, the reconstruction of the range blocks within RDBS in the first and last iterations should be performed to recover MIFS and obtain the whole decoded image, respectively. Finally, Jacquin's and three state of the art methods are adopted to verify the effectiveness of the proposed method.

References

1. B. Wohlberg and B. G. Jager, "A review of the fractal image coding literature," *IEEE Trans. on Image Process.* **8**(12), 1716-1729 (1999).
2. A. K. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. on Image Process.* **1**(1), 18-30 (1992)
3. V. Chaurasia and V. Chaurasia, "Statistical feature extraction based technique for fast fractal image compression," *J. Vis. Commun. Image R.* **41,** 87-95 (2016).
4. Y. P. Zheng, X. P. Li, and M. Sarem, "Fast fractal image compression algorithm using specific update search," *IET Image Process.* **14**(9), 1733-1739 (2020).

5. R. Gupta, D. Mehrotra, and R. K. Tyagi, "Hybrid edge-based fractal image encoding using K-NN search," *Multimed. Tools Appl.* **81,** 21135-21154 (2022).
6. F. R. Shen and H. Osamu, "A fast no search fractal image coding method," *Signal Process. Image Commun.* **19**(5), 393-404 (2004).
7. X. Y. Wang and S. G. Wang, "An improved no-search fractal image coding method based on a modified gray-level transform," *Comput. Graph.* **32**(4), 445-450 (2008).
8. X. Y. Wang, Y. X. Wang, and J. J. Yun, "An improved no-search fractal image coding method based on a fitting plane," *Image Vision Comput.* **28,** 1303-1308 (2010).
9. S. Bi and Q. Wang, "Fractal image coding based on a fitting surface" *J. Appl. Math*. **2014,** 634848 (2014).
10. M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal image denoising," *IEEE Trans. on Image Process.* **12**(12), 1560-1578 (2003).
11. M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal-wavelet image denoising revisited," *IEEE Trans. on Image Process.* **15**(9), 2669-2675 (2006).
12. J. Lu, Z. X. Ye, Y. Y. Zou, and R. S. Ye, "An enhanced fractal image denoising algorithm," *Chaos, Soliton. Fract.* **38,** 1054-1064 (2008).
13. J. H. Jeng, C. C. Tseng, and J. G. Hsieh, "Study on huber fractal image compression," *IEEE Trans. on Image Process.* **18**(5), 995-1003 (2009).
14. J. Lu, Z. X. Ye, and Y. Y. Zou, "Huber fractal image coding based on a fitting plane," *IEEE Trans. on Image Process.* **22**(1), 134-145 (2013).
15. Y. Y. Zou, H. X. Hu, J. Lu, X. X. Liu, Q. T. Jiang, and G. H. Song, "A nonlocal low-rank regularization method for fractal image coding," *Fractals.* **29**(5), 2150125 (2021).
16. C. Xu, Y. T. Ye, and Z. W. Hu, "A primal-dual algorithm for robust fractal image coding," *Fractals*. **27**(7), 1950119 (2019).
17. K. H. Chung, Y. H. Fung, and Y. H. Chan, "Image enlargement using fractal," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* 273-275 (2003).
18. C. M. Lai, K. M. Lam, Y. H. Chan, and W. C. Siu, "An efficient fractal based algorithm for image magnification," in *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing,* Hong Kong 571-574 (2004).
19. Z. P. Chen, Z. L. Ye, S. X. Wang, and G. H. Peng, "Image magnification based on similarity analogy," *Chaos, Soliton. Fract.* **40,** 2370-2375 (2009).
20. Y. C. Wee and H. J. Shin, "A novel fast fractal super resolution technique," *IEEE Trans. Consum. Electron.* **56**(3), 1537-1541 (2010).
21. Z. Hua, H. C. Zhang, and J. J. Li, "Image super resolution using fractal coding and residual network," *Complexity*. **2019,** 9419107 (2019).
22. S. M. Abdullahi, H. X. Wang, and T. Li, "Fractal coding-based robust and alignment-free fingerprint image hashing," *IEEE Trans. on Inf. Foren. Sec.* **15,** 2587-2601 (2020).

23. F. Khelaifi and H. J. He, "Perceptual image hashing based on structural fractal features of image coding and ring partition," *Multimed. Tools Appl.* **79**(27), 19025-19044 (2020).
24. M. H. Pi, M. K. Mandal, and A. Basu, "Image retrieval based on histogram of fractal parameters," *IEEE Trans. on Multimedia.* **7**(4), 597-605 (2005).
25. X. Y. Wang and Z. Chen, "A fast fractal coding in application of image retrieval," *Fractals.* **17**(4), 441-450 (2009).
26. X. Huang, Q. Zhang, and W. Liu, "A new method for image retrieval based on analyzing fractal coding characters," *J. Vis. Commun. Image R.* **24**(1), 42-47 (2013).
27. M. H. Pi, C. H. Li, and H. Li, "A novel fractal image watermarking," *IEEE Trans. on Multimedia.* **8**(3), 488-498 (2006).
28. F. Daraee and S. Mozaffari, "Watermarking in binary document images using fractal codes," *Pattern Recogn. Lett.* **35,** 120-129 (2014).
29. J. Lu, Y. Y. Zou, C. Y. Yang, and L. J. Wang, "A robust fractal color image watermarking algorithm," *Math. Probl. Eng.* **2014,** 638174 (2014).
30. C. Bisogni, M. Nappi, C. Pero, and S. Ricciardi, "FASHE: A fractal based strategy for head pose estimation," *IEEE Trans. on Image Process.* **30**, 3192-3203 (2021).
31. Bisogni, M. Nappi, C. Pero, and S. Ricciardi, "PIFS scheme for head pose estimation aimed at faster face recognition," *IEEE Trans. on Biometrics, Behavior, and Identity Science* 4, 173-184 (2022).